COMPUTATIONAL METHODS FOR CREATIVE INSPIRATION IN THEMATIC TYPOGRAPHY AND DANCE

A Dissertation
Presented to
The Academic Faculty

By

Purva Tendulkar

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
College of Computing
Department of Computer Science

Georgia Institute of Technology

August 2020

COMPUTATIONAL METHODS FOR CREATIVE INSPIRATION IN THEMATIC TYPOGRAPHY AND DANCE

Thesis committee:

Dr. Devi Parikh
Department of Computer Science
Georgia Institute of Technology

Dr. Mark Riedl
Department of Computer Science
Georgia Institute of Technology

Dr. Dhruv Batra
Department of Computer Science
Georgia Institute of Technology

Date approved: July 24th, 2020

A goal without a plan is just a wish.

Antoine de Saint-Exupéry

To Aai, Baba and Tai for always being there with me.

ACKNOWLEDGMENTS

I would like to thank my advisor, Devi Parikh, for giving me the opportunity to be a part of the wonderful Visual Intelligence Lab. It has been a privilege to work with her. She has given me a lot of freedom in choosing projects and has stood by me, guiding me when I needed help, while also giving me enough space to make my own decisions and learn from my own experiences. Her succinct writing, prompt responses, organizational skills, and emphasis on clear communication (thereby necessitating clarity of thought) are qualities which have inspired me and which I have tried to imbibe not just in my research practices but also in my personal life. For all this, I am truly grateful to Devi.

I would also like to thank Dhruv Batra for his support and feedback throughout the course of my Masters. His Deep Learning and (online) Machine Learning videos have helped me tremendously. Also, his unfiltered thoughts during lab group meetings have helped shape my approach towards research.

I would like to thank Kalpesh Krishna for introducing me to the world of research and deep learning. Kalpesh has been a dear friend and mentor to me right from my undergraduate days. He was the one who taught me how to code in Python, how to use LaTeX, git, TensorFlow and PyTorch. He got me excited about reading papers, implementing them efficiently and writing reproducible code. He motivated me and inspired me to dream of opportunities I could not have imagined for myself. Today, I would not be where I am without him.

I would like to thank my internship mentors at AiBee, Chunhui Gu, Juan Carlos Niebles, Sinisa Todorovic and Silvio Savarese for their wonderful guidance and for providing a great environment during my internship. I would also like to thank Ani Kembhavi for his remote mentorship over the last one year. A special thank you to Abhishek Das for his mentorship, honest feedback, constant support, and for being a great friend.

I would also like to thank each and every member of the Computer Vision Machine Learning

& Perception (CVMLP) Group for providing a wonderful atmosphere and a great culture during my Masters. This includes – Ramprasaath Selvaraju, Harsh Agrawal, Abhishek Das, Arjun Chandrasekaran, Yash Goyal, Aishwarya Agrawal, Ashwin Kalyan, Prithvijit Chattopadhyay, Viraj Prabhu, Deshraj Yadav, Ayush Srivastava, Samyak Datta, Rishabh Jain, Erik Wijmans, Mohit Sharma, Sameer Dharur, Karan Desai, Yash Kant, Nirbhay Modhe, Zhile Ren, Meera Hahn, Joanne Truong, Peter Anderson, Vishvak Murahari and Arjun Majumdar. The immense constructive feedback the group generally provides during group presentations helped me grow and restructure my thought process. I am deeply honored to have been a part of this CVMLP Group.

I would like to thank my roommate Shubhangi Gupta for all the positivity, support and fun memories especially during the initial part of Masters.

I would like to specially thank Ram for being a constant support system for the last 2 years. Right from making me feel comfortable in the lab group and patiently answering my silliest doubts, he has been a teacher, mentor, critic and most importantly a dear friend. I would like to thank him for all the positivity and for helping me believe in myself. Working with him has been an absolute treat and I look forward to more collaborations in the future.

I would also like to specially thank Ram, Abhishek, Varun and Yash for all the fun bonding times, the sweet memories, and for keeping me sane during this quarantine period.

I would like to thank my parents, Sujata and Milind, for all their sacrifices throughout my life for the well-being of me and my sister, Pooja, for understanding and encouraging us to pursue our dreams and for their love and constant support throughout our schooling.

TABLE OF CONTENTS

LIST OF FIGURES

SUMMARY

As progress in technology continues, there is a need to adapt and upscale tools used in artistic and creative processes. This can either take the form of generative tools which can provide inspiration to artists, human-AI co-creative tools or tools that can understand and automate time-consuming labor so that artists can focus on the creative side of their art.

This thesis aims to address two of these challenges: generating tools for inspiration and automating labor-intensive, tedious work. We approach this by attempting to create interesting art by combining the best of what humans are naturally good at – heuristics of 'good' art that an audience might find appealing – and what machines are good at – optimizing well-defined objective functions. Specifically, we introduce two tasks – 1) artistic typography given an input word and theme, and 2) dance generation given any input music. We evaluate our approaches on both these tasks and show that humans find the results generated by our approaches more creative compared to meaningful baselines. The comments received from participants in our studies reveal that they found our tasks fun and intriguing. This further motivates us to push research towards using technology for creative applications.

CHAPTER 1

INTRODUCTION AND BACKGROUND

1.1 Humans, Creativity & Technology

Creative learning is central to human development – our ability to generate or recognize ideas or possibilities, see things in new perspectives and discover unique associations in everyday life helps us adapt and improvise to the changing world. However, there are few who possess the skills to conceptualize their ideas and communicate these creative thoughts effectively – we call them artists.

Today, artists play an important role in society by changing opinions, instilling values and translating experiences across space and time. While creating art, artists often draw inspiration from their own life, by observing the world around them and looking at other art.

While science and technology are primarily targeted towards efficiency and robust solutions for the market, these technologies are not very easily accessible to creative communities possibly due to the lack of smart tools that can truly *understand* the artistic value of an invention (taking its subjectivity into account) and provide relevant feedback or recommendations. This demands the development of dedicated tools to aid artists in their creative processes. This may can range from machines automating trivial, labor-intensive work that artists may need to go through so that they may better spend their time on what matters most, i.e., the creative aspect, to machines actually generating art that may serve as inspiration.

1.2 Contributions

In this work, we describe two efforts in developing tools that can provide creative inspiration to artists and that can just be used for fun by lay people as well.

In the first work, we describe an unsupervised approach to creatively stylize a word using theme-based cliparts. More concretely given an input word and a theme, the individual letters of the input word are replaced by cliparts relevant to the theme which visually resemble the letters – adding creative context to the potentially boring input word. We define a set of evaluation metrics for this task and evaluate our approach under these metrics. We conduct human evaluation studies and show that our approach outperforms meaningful baselines in terms of word recognition, theme recognition, as well as creativity.

In the second work, we present a general computational approach that enables a machine to generate a dance for any input music. Our approach encodes intuitive, flexible heuristics of what makes a dance "good" by aligning the structure of the music with the set of dance movements. We find that this simple heuristic guides agents to automatically discover creative dances. We conduct human studies to evaluate creativity, surprise and inspiration and show that humans find our dances superior compared to meaningful baselines.

Both these works are attempts to create tools that may assist or provide context-aware recommendations to graphic designers and choreographers. They are not meant to replace, but rather augment human creations and improve with feedback from professionals.

1.3 List of Publications

- **Feel The Music: Automatically Generating A Dance For An Input Song**

  Purva Tendulkar, Abhishek Das, Ani Kembhavi, Devi Parikh

  11th International Conference on Computational Creativity (ICCC 2020)

- **SQuINTing at VQA Models: Interrogating VQA Models with Sub-Questions**

  Ramprasaath R. Selvaraju, Purva Tendulkar, Devi Parikh, Eric Horvitz, Marco Ribeiro, Besmira Nushi, Ece Kamar

  IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2020)

- **Trick or TReAT: Thematic Reinforcement for Artistic Typography**

  Purva Tendulkar, Kalpesh Krishna, Ramprasaath R. Selvaraju, Devi Parikh

  10th International Conference on Computational Creativity (ICCC 2019)

CHAPTER 2

TRICK OR TREAT : THEMATIC REINFORCEMENT FOR ARTISTIC TYPOGRAPHY

2.1 Introduction

We address the task of theme-based word typography: given a word (e.g., `exam`) and a theme (e.g., `education`), the task is to *automagically* produce a "doodle"[1] for the word in that theme as seen in Figure 2.1. Concretely, the task is to replace each letter in the input word with a clipart from the input theme to produce a doodle, such that the word and theme can be easily identified from the doodle. Solving this task would be of value to a variety of creative applications such as stylizing text in advertising, designing logos – essentially any application where a message needs to be conveyed to an audience in an effective and concise manner.

Using graphic elements to emphasize the meaning of a word in reference to a related theme is referred to in graphic design as *semantic reinforcement*. This can be achieved in a number of ways, e.g., using different fonts and colors (Figure 2.2a), changing the position of letters relative to one another (Figure 2.2b), arranging letters in a specific direction or shape (Figure 2.2c), excluding some letters (Figure 2.2d), adding icons near or around the letters (Figure 2.2e), or replacing letters with icons (Figure 2.2f). In our work, we focus on this last type, i.e., semantic reinforcement via replacement.

This is a challenging task even for humans. It not only requires domain-specific knowledge for identifying a set of relevant cliparts to choose from, but also requires creative abilities to

[1] In this work, we use "doodle" to refer to Google doodles-like typography as in Figure 2f.

[2] Unless stated otherwise, all cliparts in the paper have been taken from The Noun Project - https://thenounproject.com/. The Noun Project contains cliparts created by different graphic designers on a variety of themes.

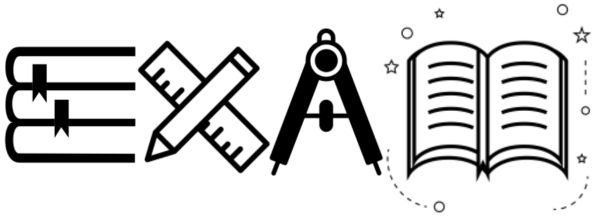


Figure 2.1: A sample doodle (that we call TReAT) generated by our system for the input word `exam` and theme `education`[2].

be able to visualize a letter in a clipart, and choose the best clipart for representing it.

The latter alone is challenging to automate – both from a training and evaluation perspective. Training a model to automatically match letters to graphics is challenging because there is a lack of large-scale text-graphic paired datasets in each domain that might be of interest (e.g., clipart, logogram). Evaluation and thus iterative development of such models is also challenging because of subjectivity and inter-human disagreement on which letter resembles which graphic.

(a) (b) (c) (d) (e) (f)

Figure 2.2: Different methods for semantic reinforcement. a) font and color variations b) positioning of letters relative to each other c) arrangement of letters in a specific shape or direction d) exclusion of some letters e) addition of icons near letters f) replacement of letters. In this work we focus on f), semantic reinforcement via replacement. [3]

We present a computational approach – Thematic Reinforcement of Artistic Typography (TReAT) – to generate doodles (TReATs) for semantic reinforcement of text via replacement. We represent letters in different fonts and cliparts from the Noun Project. in a common latent space. These latent representations are learned such that they have two characteristics: (1) The letters can be correctly recognized (e.g., *a* vs. *b*) in the latent space and (2) The letters and cliparts can be reconstructed accurately from the latent space. A reconstruction loss ensures that letters and clipart that are close in the latent space also look similar in the image space. A classification loss ensures that the latent space is informed by discriminative features that make one letter different from the other. This allows us to match cliparts to letters in a way that preserves distinctive visual features of the letters, making it easier for

humans to identify the letter being depicted by a clipart. At test time, given a word and a theme as input, we first retrieve cliparts from the Noun Project that match that theme. For each letter in the word, we find the theme-relevant clipart which minimizes the distance from it across a variety of fonts. If the distance is low enough, we replace the letter with the clipart.

We run human studies to show that subjects can reliably recognize the word as well as the theme from our TReATs, and find them to be creative. We consider a TReAT to be creative if it is surprising and/or intriguing and/or fun.

2.2 Related work

Early human communication was through symbols and hieroglyphs [1], [2]. This involved the use of characters to represent an entire word, phrase or concept. Then language evolved and we started using the alphabet for creation of new words to represent concepts. However many languages (e.g. Chinese and Japanese) still make use of pictograms or logograms to depict specific words. Today, symbols and logos are used for creative applications to increase the communication bandwidth – to convey abstract concepts, express rich emotions, or reinforce messages [3], [4], [5]. Our work produces a visual depiction of text by reasoning about similarity between the visual appearance of a letter and clipart imagery. We describe prior work in each of these domains: creativity through imagery and creativity through visual appearance of text.

2.2.1 Creativity through imagery

There has been work on evoking emotional responses through the modification of images. Work on visual blending of emojis combines different concepts to create novel emojis [6]. Visual blending has also been explored for combining two animals to create images depicting

[3]Examples a) to e) were taken from this answer on StackExchange. Example f) is a Google Doodle.

fictional hybrid animals [7]. Our approach tries to induce creativity by entirely replacing a letter with a clipart.

Work on Vismantic [8] represents abstract concepts visually by combining images using juxtaposition, fusion, and replacement. Our work also represents a theme via replacement (replacing letters with cliparts); however our replacement is for the purposes of lexical resolution, not visual. Recently, there has been an exploration of neural style transfer for logo generation [9]. This work however only transfers color and texture from the style to the content. In our work, the transfer occurs via direct replacement. Logo generation has also been explored through the use of Generative Adverserial Networks [10].

Recently Google's QuickDraw! and AutoDraw based on sketch-rnn [11] have gained a lot popularity. Their work trains a recurrent neural network (RNN) to construct stroke-based drawings of common objects, and is also able to recognize objects from human strokes. One could envision creating a doodle by writing out one letter at a time, that AutoDraw would match to the closest object in its library. However, these matches would not be theme based. Iconary [4] is a very recent pictionary-like game that users can play with an AI. Relevant to this work, user drawings in Iconary are mapped to icons from The Noun Project to create a scene.

The use of conditional adversarial networks for image-to-image translation is gaining popularity. However, using a pix2pix-like architecture [12] for our task would involve the use of labeled pairwise (clipart, letter) data, which as discussed earlier, is hard to obtain. CycleGAN [13] does not require paired label data, but is not a good fit for our task because we are interested in matching letters to cliparts from a specific theme. The pool of clipart is thus limited, and would not be sufficient to learn the target domain. Finally, generative modeling is typically lossy; we prefer direct replacement of cliparts for greater readability.

[4] https://iconary.allenai.org/

2.2.2 Creativity through visual appearance of text

Advances in conditional Generative Adverserial Networks (cGANs) [14] have motivated style transfer for fonts [15] through few-shot learning. Work on learning a manifold of fonts [16] allows everyday users to create and edit fonts by smoothly interpolating between existing fonts. [17] explore an evolutionary system for the generation of type stencils constructed from line segments. The first work explores the creation of unseen letters of a known font, while the other two works explore the creation of entirely new fonts – neither add any theme-related semantics or additional graphic elements to the text.

Work on neural font style transfer between fonts [18] explores the effects of using different weighted factors, character placements and orientations in style transfer. This work also has an experiment using icons as style images, however the style transfer is only within the context of visual features of icons such as the texture and thickness of strokes, as opposed to direct replacement.

MarkMaker [5] generates logos based on company names – primarily displaying the name in various fonts and styles, sometimes along with a clipart. It uses a genetic algorithm to iteratively refine suggestions based on user feedback.

2.3 Approach

In this section, we first describe our procedure for collecting training data, and then our model and its training details. Finally, we describe our test-time procedure to generate a TReAT, that is, obtaining theme-based clipart matches for an input word and theme. A sketch of our model along with examples for training and testing are shown in Figure 2.3.

[5] https://emblemmatic.org/markmaker/

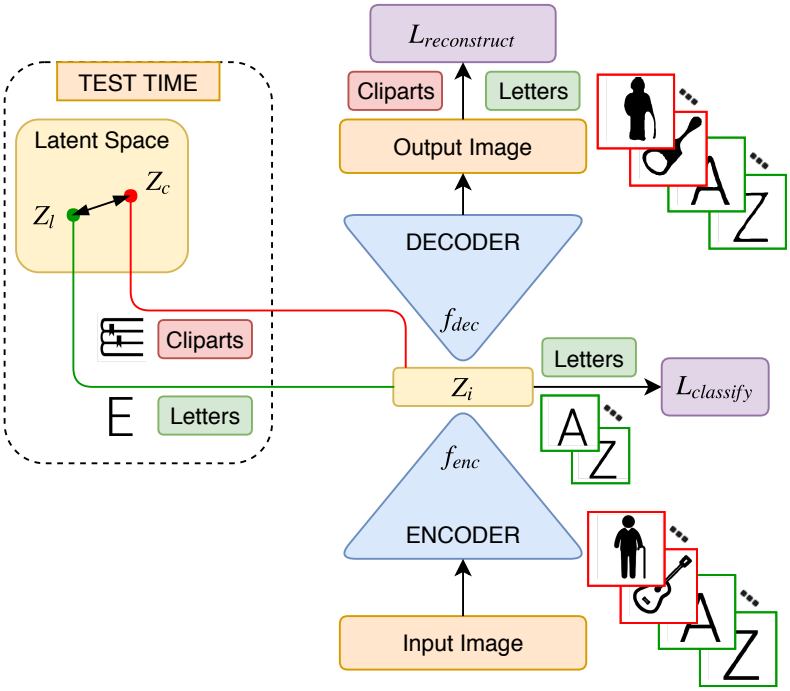


Figure 2.3: Block diagram describing our model during training and testing. The model is trained on a reconstruction and classification loss in a multitask fashion. During inference, latent space distances are calculated to match letters to cliparts. See text for more details.

2.3.1 Training Data

For our task we need two types of data for training – letters in different fonts, and cliparts. Note that we do not need a *correspondence* between the letters and cliparts. In that sense, as stated earlier, our approach is an unsupervised one.

For clipart data, we use the Noun Project – a website that aggregates and categorizes symbols that are created and uploaded by graphic designers around the world. The Noun Project cliparts are all 200 $\times$ 200 in PNG format. The Noun Project has binary cliparts, and will result in TReATs of the style shown in Figure 2.1. Different choices of the source of cliparts can result in different styles, including colored TReATs similar to Figure 2.2f. We downloaded a random set of $\sim$50k cliparts from the Noun Project.

We obtain our letter data from a collection of 1400 distinct font files. [6] On manual inspection, we found that this set contained a lot of visual redundancies (e.g. the same

[6] These font files (TTF) were obtained from a designer colleague.

Figure 2.4: Example TReATs generated by our approach for (word & theme) pairs: a) (`canoe` & `watersports`) b) (`world` & `countries, continents, natural wonders`) c) (`water` & `drinks`) d) (`church` & `priest, nun, bishop`)

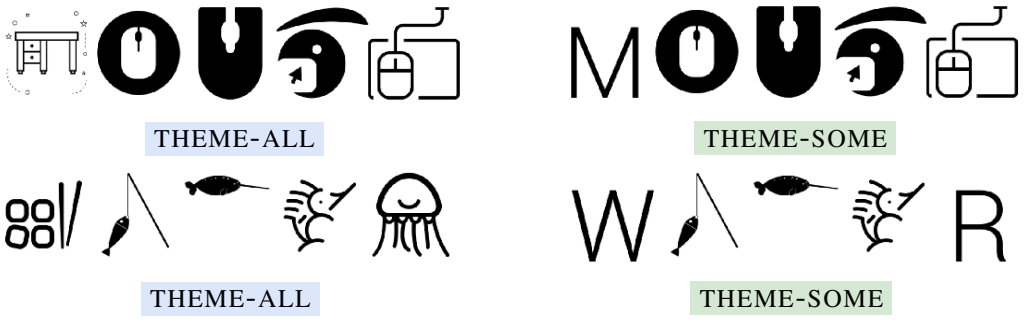


Figure 2.5: We replace letters in a word with cliparts only if the clipart is sufficiently similar to the letter, placing more stringent conditions on the first and last letters in the word. Notice that in each pair, the TReATs on the right (with a subset of letters replaced) are more legible (`Mouse` and `Water`) than the ones on the left, while still depicting the associated themes (`computer` and `fish, mermaid, sailor`).

font being repeated in regular and bold weight types). We removed such repetitions. We also manually inspected the data to ensure that the individual letters were recognizable in isolation, and discarded overly complicated and intricate font styles. This left us with a total of 777 distinct fonts. We generated 200 $\times$ 200 image files (PNG format) from each font file for the entire alphabet (uppercase and lowercase) giving us a total of 40.4k images of letters (777 fonts $\times$ 26 letters in the English alphabet $\times$ 2 (upper and lower cases)).

2.3.2 Model

Our primary objective is to find visual similarities between cliparts and letters in an unsupervised manner. To this end, we train an autoencoder [19] with a reconstruction loss on both clipart and letter images (denoted by $\mathbf{X}_{cl}$). We denote a single input image by X_i. Each

input image X_i is passed through an encoder neural network $f_{\text{enc}}(\cdot)$ and projected to a low dimensional intermediate representation Z_i. Finally, a decoder neural network $f_{\text{dec}}(\cdot)$ tries to reconstruct the input image as $\hat{X}_i$, using the objective $\mathcal{L}_{\text{reconstruct}}$,

$$Z_i = f_{\text{enc}}(X_i) \tag{2.1}$$

$$\hat{X}_i = f_{\text{dec}}(Z_i) \tag{2.2}$$

$$\mathcal{L}_{\text{reconstruct}} = \frac{1}{|\mathbf{X}_{cl}|} \sum_{i \in \mathbf{X}_{cl}} \text{SSD}(X_i, \hat{X}_i) \tag{2.3}$$

where $\text{SSD}(X_i, \hat{X}_i)$ is the sum over squared pixel differences between the original image and its reconstruction. We set the dimensionality of Z_i to be 128. In addition to the reconstruction objective, we utilize letter labels (52 labels for lowercase and uppercase letters) to classify the intermediate representations Z_i for the letter images. This objective helps the encoder discriminate between different letters (possibly with similar visual features) while clustering together the intermediate representations for the same letter across different fonts. This would allow the intermediate representation to capture visual features that are characteristic of each letter, and when cliparts are matched to letters using this representation, the matched cliparts will retain the visually discriminative features of letters.

Concretely, we project Z_i to a 52-dimensional space using a single linear layer with a softmax non-linearity and use the cross entropy loss function. Let W and b be the parameters of a linear transformation of Z_i. We obtain a probability distribution $P_i(\cdot)$ across all labels as,

$$P_i(\cdot) = \text{softmax}(WZ_i + b) \tag{2.4}$$

Let $\mathbf{X}_l$ be the subset of images in $\mathbf{X}_{cl}$ that are letters. We maximize the probability of the correct label Y_i corresponding to each letter image X_i.

$$\mathcal{L}_{\text{classify}} = -\frac{1}{|\mathbf{X}_l|} \sum_{i \in \mathbf{X}_l} \log P_i(Y_i) \tag{2.5}$$

Note that the same Z_i is used in both objective functions for letter images. These objectives are jointly trained using a multitask objective

$$\mathcal{L} = \alpha \mathcal{L}_{\text{reconstruct}} + (1 - \alpha)\mathcal{L}_{\text{classify}} \tag{2.6}$$

Our final loss function is thus composed of two different loss functions: (1) $\mathcal{L}_{\text{reconstruct}}$ trained on both letters and cliparts, and (2) $\mathcal{L}_{\text{classify}}$ trained only on letters. Here α is a tunable hyperparameter in the range $[0, 1]$. We set α to 0.25 after manually inspecting outputs of a few word-theme pairs we used while developing our model (different from the word-theme pairs we use to evaluate our approach later).

Implementation Details:

Our encoder network $f_{\text{enc}}(\cdot)$ is an AlexNet [20] convolutional neural network trained from scratch, made up of 5 convolutional and 3 fully connected layers. Our decoder network f_{dec} consists of 5 deconvolutional layers, 3 fully connected layers and 3 upsampling layers. We use batch norm between layers.[7] We use ReLU activations for both the encoder and decoder. We use the Adam optimizer with a learning rate of 10^{-4}, and a weight decay of 10^{-5}. The input dataset is divided into minibatches of size 100 with a mixture of clipart and letter images in each minibatch. We use early stopping based on a validation set as our stopping criterion.

Data Preprocessing:

We resize our images to 224×224 using bilinear interpolation to match the input size of our AlexNet-based encoder. We normalize every channel of our input data to fall in $[-1, 1]$.

[7] Implementations of our encoder and decoder were adapted from https://github.com/arnaghosh/Auto-Encoder.

2.3.3 Finding Matches

At test time given a word and a theme, we retrieve a theme-relevant pool of cliparts (denoted by $\mathbf{C}$) by querying Noun Project. These theme-relevant cliparts may or may not be part of the randomly downloaded $\sim$50k cliparts used during training. If multiple phrases have been used to describe a theme, we use each phrase separately as a query. We limit this retrieval to no more than 10,000 cliparts for each phrase. We then combine cliparts for different phrases of a theme together to form the final pool of cliparts for that theme. For example, for the theme `countries, continents, natural wonders`, we query Noun Project for `countries`, `continents` and `natural wonders` individually and combine all retrieved cliparts together to form the final pool of theme-relevant cliparts. On average across 95 themes we experimented with, we had a minimum of 49 and maximum of 29,580 cliparts per theme, with a mean of 9731.2 and median of 9966. We augmented this set of cliparts with left-right (mirror) flips of the cliparts. This improves the overall match quality. E.g., in cases where there existed a good match for `J`, but not for `L`, the clipart match for `J`, when flipped, served as a good match for `L`. Similarly for `S` and `Z`.

For each letter l of the input word, we choose the corresponding letter images (denoted by $\mathbf{F}_l$) taken from a predefined pool of fonts. To create the pool of letter images $\mathbf{F}_l$, we used uppercase letters from 14 distinct, readable fonts from among the 777 fonts used during training. These were kept fixed for all experiments. We found that uppercase letters had better matches with the cliparts (lower cosine distance between corresponding latent representations Z_i on average, and visually better matches). Moreover, we found that in several fonts, letters were the same for both cases.

We replace the letter with a clipart (chosen from $\mathbf{C}$) whose mean cosine distance in the intermediate latent space is the least, when computed against every letter image in $\mathbf{F}_l$. Concretely, if Z_i denotes the intermediate representation for the i^{th} image in $\mathbf{F}_l$ and Z_c is the intermediate representation for a clipart c in $\mathbf{C}$, the chosen clipart $\hat{c}_l$ is

$$\hat{c}_l = \underset{c \in \mathbf{C}}{\operatorname{argmin}} \frac{1}{|\mathbf{F}_l|} \sum_{i \in \mathbf{F}_l} \left(1 - \frac{Z_i \cdot Z_c}{|Z_i||Z_c|}\right) \tag{2.7}$$

We find a clipart that is most similar to the letter on average across fonts to ensure a more robust match than considering a single most similar font. In this way, each letter in the input word is replaced by its closest clipart to generate a TReAT.

We show example TReATs generated by our approach in Figure 2.4. We find that the word can often be difficult to recognize from the TReAT if the Noun Project cliparts corresponding to a theme are not sufficiently similar to the letters in the word. To improve the legibility of our TReATs, we first normalize the cosine distance values of our matched cliparts for the alphabet for a specific theme in the range [0, 1]. We only replace a letter with its clipart match if the normalized cosine distance between the embedding of the letter and clipart is < 0.75. It is known that the first and last letters of a word play a crucial role in whether humans can recognize the word at a glance. [21] So we use a stricter threshold, and replace the first and last letters of a word with a clipart only if the normalized cosine distance between the two is < 0.45. Example TReATs with all letters replaced and only a subset of letters replaced can be seen in Figure 2.5. Clearly, the TReATs with a subset of letters replaced (THEME-SOME) are more legible than replacing all letters (THEME-ALL), while still depicting the desired theme. We quantitatively evaluate this in the next section.

2.4 Evaluation

We evaluate our entire system along three dimensions:

- How well is our model able to learn a representation that captures visual features of the letters?
- How does our chosen source of cliparts (Noun Project) affect the quality of matches?
- How good are our generated TReATs?

Figure 2.6: We evaluate five different approaches for generating TReATs. See text for details.

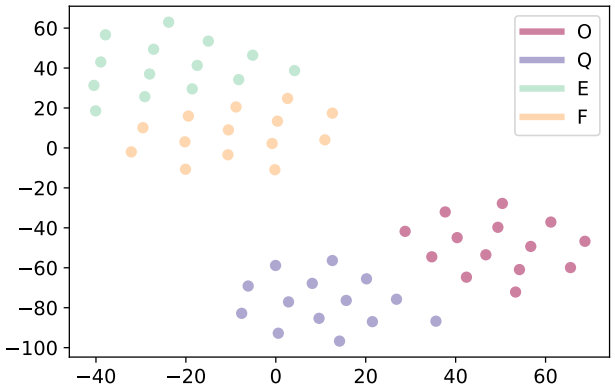


Figure 2.7: t-SNE plot showing clusters of uppercase O, Q, E and F. Each letter forms its own cluster and visually similar pairs (E & F, O & Q) form super-clusters. However, these super-clusters are far apart from each other due to significant visual differences.

2.4.1 Learnt Representation

We use t-Distributed Stochastic Neighbor Embedding (t-SNE) to visualize our learnt latent representations of letters and cliparts. Among letters, we find that our model clusters letters of different fonts together, while distinguishing between visually dissimilar letters. E.g., Figure 2.7 visualizes in 2D uppercase O, Q, E and F in the 14 fonts used at test time. As expected, O and Q clusters are close, and E and F clusters are close, but both these sets of clusters are apart. Visualizing letters as well as cliparts, Figure 2.8 shows that our model is able to learn a representation such that visually similar letter-clipart pairs are close in the latent space.

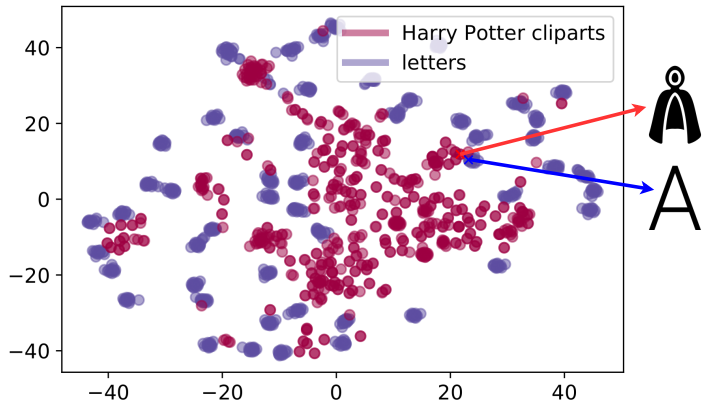
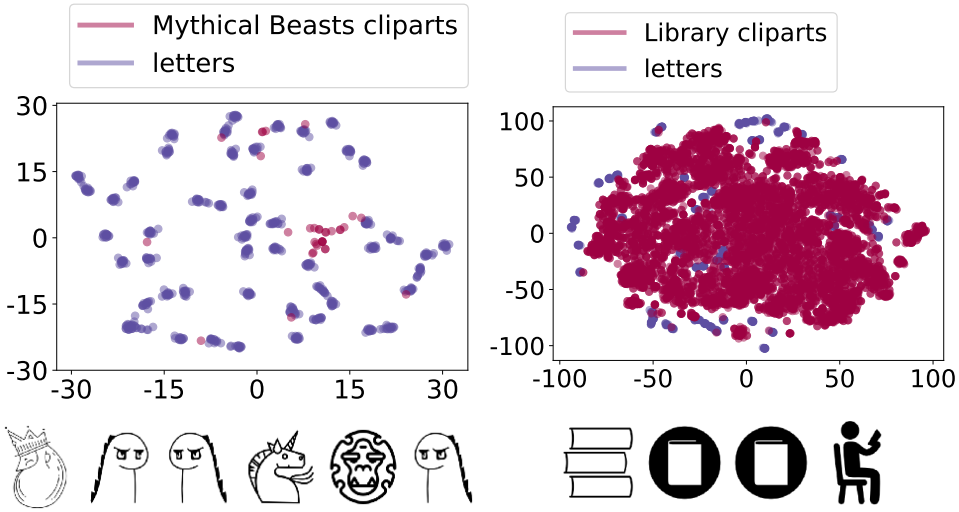


Figure 2.8: t-SNE plot showing clusters of `Harry Potter` themed cliparts along with letters. Cliparts which look like `A` lie close to the cluster of `A`'s in the latent space.



(a) (b)

Figure 2.9: Impact of diversity of cliparts from different themes in the Noun Project on corresponding TReATs. a) TReAT of `dragon` in theme `mythical beast` is not legible due to lower coverage of letters by the themed cliparts compared to a TReAT of `book` in theme `library` shown in b).

2.4.2 Effect of source of cliparts

Themes which have fewer cliparts, and hence typically lower diversity and coverage across the letters (e.g. `mythical beast` in Figure 2.9a) have poorer matches as compared to larger, more diverse themes (e.g. `library` in Figure 2.9b). Indeed, we see that recognizing the word in a TReAT generated from the former theme is significantly harder than for the latter. Note that the diversity of cliparts is more important than the quantity. Fewer but more diverse cliparts can lead to better TReATs than many but less diverse cliparts.

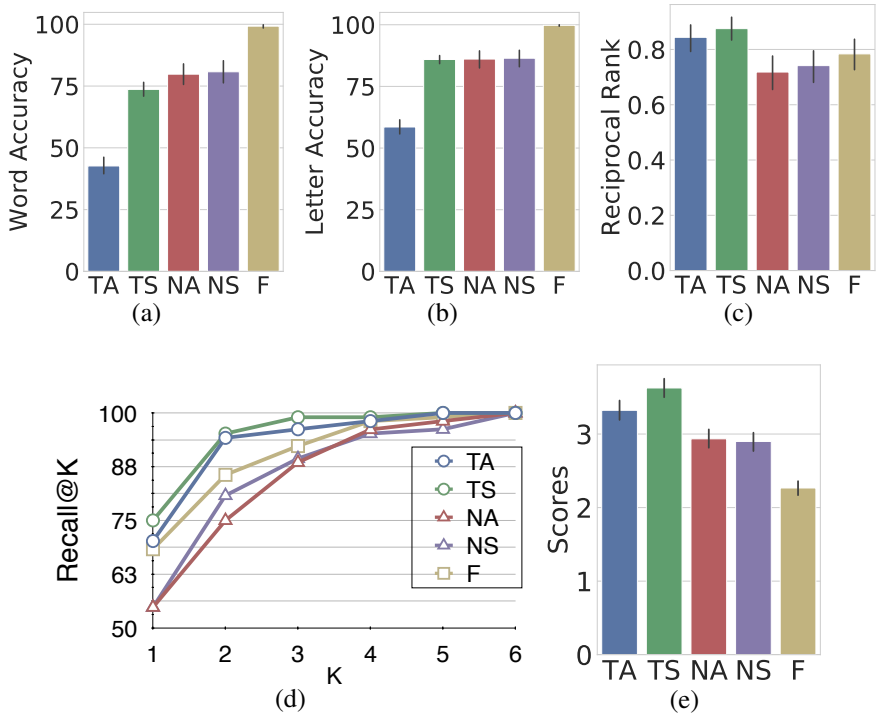


Figure 2.10: Evaluation of TReATs from five approaches (THEME-ALL (TA), THEME-SOME (TS), NOTHEME-ALL (NA), NOTHEME-SOME (NS), FONT (F)) for a) word recognition; b) letter recognition; c) and d) theme recognition; e) creativity.

2.4.3 Quality of TReATs

We now evaluate the quality of TReATs generated by our approach. We developed our approach on a few themes (e.g., `education`, `Harry Potter`, `Halloween`, `Olympics`) and associated words (e.g., `exam`, `always`, `witch`, `play`). We select these arbitrarily as diverse and popular domains. To evaluate our approach in the open world, we collected 104 word-theme pairs from subjects on Amazon Mechanical Turk (AMT). We told subjects that given a word and an associated theme, we have a bot that can draw a doodle. We showed subjects a few example TReATs. We asked subjects to give us a word and an associated theme (to be described in 1-5 comma separated phrases) that they would like to see a doodle for. Example (word & theme) pairs from our dataset are (`environment` & `pollution, dirt, wastage`), (`border` & `USA`), (`computer` & `technology`). We allowed subjects to use multiple phrases to describe the theme to allow for a more diverse set of cliparts to search from when generating the TReAT. We evaluate our TReATs

along three dimensions: 1) Can subjects recognize the word in the TReAT? 2) Can subjects recognize the theme in the TReAT? and 3) Do subjects find the TReAT creative? We conduct independent studies for each of these to eliminate the influence of one on the other.

We compare our approach THEME-SOME to a version where we replace all letters in the word with cliparts (THEME-ALL) to evaluate how replacing a subset of letters affects word recognition (expected to increase) and theme recognition (expected to remain unchanged or even increase because recognizing the word can aid in recognizing the theme), as well as creativity (expected to remain unchanged or even increase because the associated word is more legible as opposed to gibberish). We also compare our approach to an approach that replaces letters with cliparts, but is not constrained by the theme of interest (NOTHEME-SOME and NOTHEME-ALL). We find the clipart that is closest across all 95[8] themes in our dataset to replace the letter. This can result in increased word recognition because letters can find a clipart that is more similar (from a larger pool not constrained by the theme), but will result in lower theme recognition accuracy. Note that theme recognition will still likely be higher than chance because the word itself gives cues about the theme. For no-themed clipart, we compare an approach that replaces all letters (i.e., NOTHEME-ALL) as well as only a subset of letters (NOTHEME-SOME). Finally, as a point of reference, we evaluate a TReAT that simply displays the word in a slightly atypical font (FONT). We expect word recognition to be nearly perfect, but theme recognition as well as creativity to be poor. These five different types of TReATs are shown in Figure 2.6. This gives us a total of 520 TReATs to evaluate (5 types $\times$ 104 word-theme input pairs). No AMT workers were repeated across any of these tasks.

Word recognition:

We showed each TReAT to 5 subjects on AMT. 461 unique subjects participated in the word recognition study. They were asked to type out the word they see in the TReAT in free-form

[8] 95 since some themes repeat in our 104 (word & theme) pairs.

text. Notice the open-ended nature of the task. Performance of crowd-workers for word recognition of different types of TReATs is shown in Figure 2.10a. This checks for exact string matching (case-insensitive) between the word entered by subjects and the true word. As a less stringent evaluation, we also compute individual letter recognition accuracy. These were computed only for cases where the length of the word entered by the subject matched the true length of the TReAT because if the lengths do not match, the worker likely made a mistake or was distracted. Letter recognition accuracies are shown in Figure 2.10b.

As expected, leaving a subset of the letters unchanged leads to a higher recognition rate for THEME-SOME and NOTHEME-SOME compared to their counterparts, THEME-ALL and NOTHEME-ALL respectively. Also, NOTHEME-ALL and NOTHEME-SOME have higher word recognition accuracy than THEME-ALL and THEME-SOME because the clipart matches are obtained from a larger pool (across all themes rather than from a specific theme). The added signal from the theme of the cliparts in THEME-ALL and THEME-SOME does not help word recognition enough to counter this. NOTHEME-ALL already has a high recognition rate, leaving little scope for improvement for NOTHEME-SOME. Finally, FONT has near perfect word recognition accuracy because it contains the word clearly written out. It is not a 100% because of typos on the part of the subjects. In some cases we found that subjects did not read the instructions and wrote out the theme instead of the word itself across all TReATs. These subjects were excluded from our analysis.

Theme recognition:

We showed each TReAT to 6 subjects on AMT. 163 unique subjects participated in the theme recognition study. The same theme can be described in many different ways. So unlike word recognition, this task could not be open-ended. For each TReAT, we gave subjects 6 themes as options from which the correct theme is to be identified. These 6 options included the true theme from the 95 themes in our dataset, 2 similar themes, and 3 random themes. The

similar themes are the 2 nearest neighbor themes to the true theme in word2vec [22] space. word2vec is a popular technique to generate vector representations of a word or "word embeddings" which capture the meaning of the word such that words that share common contexts in language (that is, likely have similar meaning) are located in close proximity to one another in the space. If a theme is described by multiple words, we represent the theme using the average word2vec embedding of each word. This is a strategy that is commonly employed in natural language processing to reason about similarities between phrases or even entire sentences [23], [24]. For example, the options for Figure 2.4c were 1) home, furniture, 2) drinks, 3) corpse, undertaker, vampire, 4) food, dessert, 5) birds and 6) food with the correct answer being drinks.

We find that 64% of the TReATs were assigned to the correct theme for THEME-ALL, 67% for THEME-SOME, 43% for NOTHEME-ALL, 51% for NOTHEME-SOME and 60% for FONT respectively. As expected, NOTHEME-ALL and NOTHEME-SOME have lower theme recognition accuracy than THEME-ALL and THEME-SOME because NOTHEME-ALL and NOTHEME-SOME do not use cliparts from specific themes. Notice that theme recognition accuracy is still quite high, because the word itself often gives away cues about the theme (as seen by the theme recognition accuracy of FONT that lists the word without any clipart).

This theme recognition rate is a pessimistic estimate because theme options presented to subjects included nearest neighbors to the true theme as distractors. These themes are often synonymous to the true theme. As a less stringent evaluation, we sort the 6 options for each TReAT based on the number of votes the option got across subjects. Figure 2.10c shows the Mean Reciprocal Rank of the true option in this list (higher is better). We also show Recall@K in Figure 2.10d that compute how often the true option is in the top-K in this sorted list. Similar trends as described above hold.

Comparing THEME-SOME to THEME-ALL, we see that replacing only a subset of letters does not hurt theme recognition (in fact, it improves slightly), but improves word

recognition significantly. So overall, THEME-SOME produces the best TReATs. We see this being played out when TReATs are evaluated for their overall creativity (next). This relates to Schmidhuber's theory of creativity [25]. He argues that data is creative if it exhibits both a learnable or recognizable pattern (and is hence compressible), and novelty. THEME-SOME achieves this balance.

Creativity:

Recall that our goal here is to create TReATs to depict words with visual elements such that the TReAT leaves an impression on people's minds. We now attempt to evaluate this. Do subjects find the TReAT intriguing / surprising / fun (i.e., creative)? We showed each TReAT to 5 subjects on AMT. 207 unique subjects participated in the creativity study. They were told: "This is a doodle of `[word]` in a `[theme]` theme. On a scale of 1-5, how much do you agree with this statement? This doodle is creative (i.e, surprising and/or intriguing and/or fun). a. Strongly agree (with a grin-like smiley face emoji in green) b. Somewhat agree (with a smiley face in lime green) c. Neutral (with a neutral face in yellow) d. Somewhat disagree (with a slightly frowning face in orange) e. Strongly disagree (with a frowning face in red)." The associated scores were 5 to 1 respectively. Crowd-worker scores are shown in Figure 2.10e. THEME-SOME was rated the highest. We believe this is due to a good trade off between legibility and having a theme-relevant depiction that allows for semantic reinforcement. NOTHEME-ALL and NOTHEME-SOME are significantly worse. Recall that they are visual, but not in a theme-specific way. So they are visually interesting, but do not allow for semantic reinforcement. The resultant reduction in creativity is evident. Interestingly, NOTHEME-SOME scores slightly higher than NOTHEME-ALL. This may be because NOTHEME-SOME is not more legible than NOTHEME-ALL (NOTHEME-ALL is already sufficiently legible). With more of the letters visually depicted, NOTHEME-ALL is more interesting. Finally, FONT has a significantly lower creativity score. It is rated lower than neutral, close to the "Somewhat disagree" rating. To get a qualitative sense, we asked

subjects to comment on what they think of the TReATs. Some example comments:

THEME-ALL : *"cool characters and each one fits the theme of the ocean"*, *"Its [sic] creative and represents the theme well, but I don't see disney all that much."*

THEME-SOME : *"I like how it uses the image of the US and then a state to spell out the word and looks like something you'd remember."*, *"Very fun and intriguing. I like how all the letters are pictures representing a computer mouse."*.

NOTHEME-ALL : *"It is creative but it has nothing to do with fear."*, *"It does a very good job of spelling out CHRISTMAS, but the individual letters are not related to the holiday at all."*

NOTHEME-SOME : *"It is somewhat creative, especially the unicorn head for "G", though I don't know what any of it has to do with the theme."*, *"There are too many icons that seemingly have nothing to do with the theme."*

FONT : *"It just spells out the word, not really a doodle"*, *"Its not a doodle, its just the word Parrot, so I don't think its creative at all."*

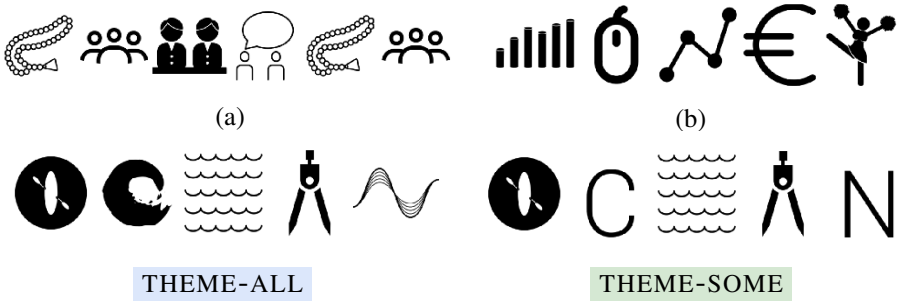


Figure 2.11: Example failure modes of our approach – a) more theme-relevant icons such as the cross should be used to depict the theme `pastor, Jesus, people, steeple`; b) lack of context wherein `support` here actually refers to *financial* support, and not the motivational support which comes from a cheerleader depicting `y` in the word `money`; Bottom: `C` and `N` are replaced in our final THEME-SOME TReAT even when the matches were actually quite relevant and visually fitting.

2.5 Future Work

In this section, we discuss some drawbacks of our current model and potential future work.

No Clipart Relevance Score:

A comment from a subject evaluating the creativity of Figure 2.11a (word `church` & theme `pastor, Jesus, people, steeple`) was *"you need a cross [...] before the general public would [...] get this."* Our approach does not include world knowledge that indicates which symbols are canonical for themes (Noun Project does not provide a relevance score). As a result, our model can not explicitly trade off visual similarity (VS) for theme relevance (TR) – either to compromise on VS to improve TR, or to at least optimize for TR if VS is poor.

No Contextual Querying:

Multiple phrases used to describe a theme often lose context when they are individually queried into the Noun Project. For example, the clipart in Figure 2.11b for (word `money` & theme `finance, banking, support`) contains the image of a cheerleader, which is relevant to the phrase `support`, but is not relevant in the context of the `finance` theme.

The lack of context also hurts polysemous theme words. `bat` when used as a keyword with the another keyword `bird` refers to the creature bat, but in the context of `baseball` refers to sports equipment.

Imperfect Match Scores:

Our automatic similarity score frequently disagrees with our (human) perceptual notion of similarity. E.g., in Figure 2.11 right, the cliparts used to replace `C` and `N` in THEME-ALL look sufficiently similar to the corresponding letters. But the automatic similarity score was low, and so THEME-SOME chose to not replace the letters. Approaches to improve the automatic score can be explored in future work. For instance, in addition to mirror images, using rotated and scaled versions of the cliparts to augment the dataset would help.

2.5.1 Interactive Interface

To mitigate these concerns, we plan to build an interactive tool. Users can choose from the top-k clipart matches for each letter. Users can iterate on the input theme descriptions until they are satisfied with the TReAT. Users can also leave the theme unspecified in which case we can use the word itself as the theme. Finally, users can choose which letters to replace in THEME-SOME like TReATs.

2.6 Conclusion

In this work, we introduce a computational approach for semantic reinforcement called TReAT – Thematic Reinforcement for Artistic Typography. Given an input word and a theme, our model generates a "doodle" (TReAT) for that word using cliparts associated with that theme. We evaluate our TReATs for word recognition (can a subject recognize the word being depicted?), theme recognition (can a subject recognize what theme is being illustrated in the TReAT?), and creativity (overall, do subjects find the TReATs surprising / intriguing / fun?). We find that subjects can recognize the word in our TReATs 74% of the time, can recognize the theme 67% of the time, and on average "Somewhat agree" that our TReATs are creative.

CHAPTER 3

FEEL THE MUSIC: AUTOMATICALLY GENERATING A DANCE FOR AN INPUT SONG

3.1 Introduction

Dance is ubiquitous human behavior, dating back to at least $20,000$ years ago [26], and embodies human self-expression and creativity. At an 'algorithmic' level of abstraction, dance involves body movements organized into spatial patterns synchronized with temporal patterns in musical rhythms. Yet our understanding of how humans represent music and how these representations interact with body movements is limited [27], and computational approaches to it under-explored.

We focus on automatically generating creative dances for a variety of music. Systems that can automatically recommend and evaluate dances for a given input song can aid choreographers in creating compelling dance routines, inspire amateurs by suggesting creative moves, and propose modifications to improve dances humans come up with. Dancing can also be an entertainment feature in household robots, much like the delightful ability of today's voice assistants to tell jokes or sing nursery rhymes to kids!

Automatically generating dance is challenging for several reasons. First, like other art forms, dance is subjective, which makes it hard to computationally model and evaluate. Second, generating dance routines involves synchronization between past, present and future movements whilst also synchronizing these movements with music. And finally, compelling dance recommendations should not just align movements to music, they should ensure these are enjoyable, creative, and appropriate to the music genre.

As a step in this direction, we consider simple agents characterized by a single movement

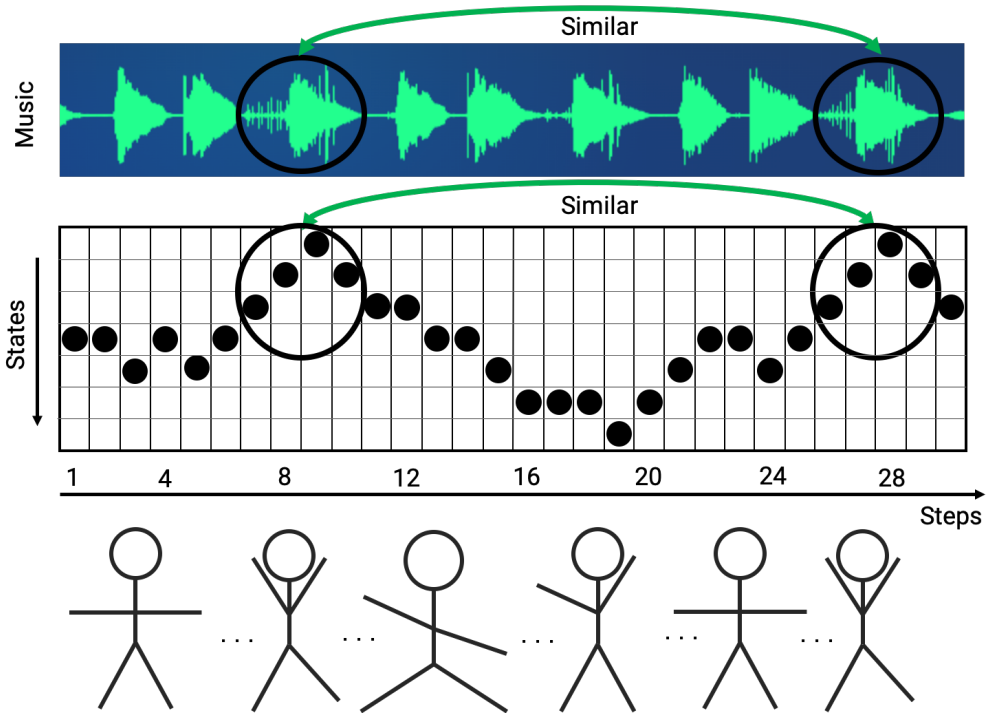


Figure 3.1: Given input music (top), we generate an aligned dance choreography as a sequence of discrete states (middle) which can map to a variety of visualizations (e.g., humanoid stick-figure pose variations, bottom). Video available at https://tinyurl.com/ybfakpxf.

parameter that takes discrete ordinal values. Note that a variety of creative visualizations can be parameterized by a single value, including an agent moving along a 1D grid, a pulsating disc, deforming geometric shapes, or a humanoid in a variety of sequential poses.

In this work, we focus on designing interesting choreographies by combining the best of what humans are naturally good at – heuristics of 'good' dance that an audience might find appealing – and what machines are good at – optimizing well-defined objective functions. Our intuition is that in order for a dance to go well with the music, the overall spatio-temporal movement pattern should match the overall structure of music. That is, if the music is similar at two points in time, we would want the corresponding movements to be similar as well (Fig. Figure 3.1). We translate this intuition to an objective our agents optimize. Note that this is a flexible objective; it does not put constraints on the specific movements allowed. So there are multiple ways to dance to a music such that movements at points in time are similar when the music is similar, leaving room for discovery of novel dances. We experiment with 25 music clips from 13 diverse genres. Our studies show that human subjects find our

dances to be more creative compared to meaningful baselines.

3.2 Related work

Music representation. [28, 29] use beat timings and loudness as music features. We use Mel-Frequency Cepstral Coefficients (MFCCs) that capture fine-grained musical information. [30] use the power spectrum (FFT) to represent music. MFCC features better match the exponential manner in which humans perceive pitch, while FFT has a linear resolution.

Expert supervision. Hidden Markov Models have been used to choose suitable movements for a humanoid robot to dance to a musical rhythm [31]. [32, 33, 34] trained stick figures to dance by mapping music to human dance poses using neural networks. [35] trains models on human movement to generate novel choreography. Interactive and co-creation systems for choreography include [36, 37]. In contrast to these works, our approach does not require any expert supervision or input.

Dance evaluation. [38] evaluate generated dance by asking users whether it matches the "ground truth" dance. This does not allow for creative variations in the dance. [32, 34] evaluate their dances by asking subjects to compare a pair of dances based on beat rate, realism (independent of music), etc. Our evaluation focuses on whether human subjects find our generated dances to be creative and inspiring.

3.3 Dataset

For most of our experiments, we created a dataset by sampling $\sim$10-second snippets from 22 songs for a total of 25 snippets. We also show qualitative results for longer snippets towards the end of the paper. To demonstrate the generality of our approach, we tried to ensure our dataset is as diverse as possible: our songs are sampled from 1) 13 different genres: Acapella, African, American Pop, Bollywood, Chinese, Indian-classical, Instrumental, Jazz,

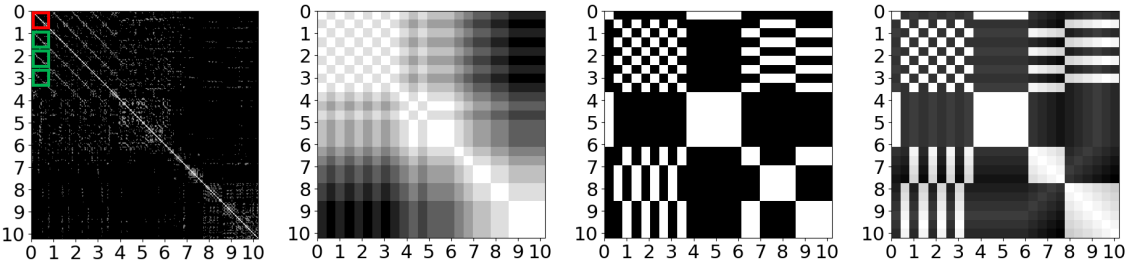


Figure 3.2: Music representation (left) along with three dance representations for a well-aligned dance: state based (ST), action based (AC), state and action based (SA).

Latin, Non-lyrical, Offbeat, Rap, Rock 'n Roll, and have significant variance in 2) number of beats: from complicated beats of Indian-classical dance of Bharatnatyam to highly rhythmic Latin Salsa 3) tempo: from slow, soothing Sitar music to more upbeat Western Pop music 4) complexity (in number and type of instruments): from African folk music to Chinese classical 5) and lyrics (with and without).

3.4 Approach

Our approach has four components – the music representation, the movement or dance representation (to be aligned with the music), an alignment score, and our greedy search procedure used to optimize this alignment score.

Music representation. We extract Mel-Frequency Cepstral Coefficients (MFCCs) for each song. MFCCs are amplitudes of the power spectrum of the audio signal in Mel-frequency domain. Our implementation uses the Librosa library [39]. We use a sampling rate of 22050 and hop length of 512. Our music representation is a self-similarity matrix of the MFCC features, wherein $\text{music}[i, j] = \exp\left(-||\text{mfcc}_i - \text{mfcc}_j||_2\right)$ measures how similar frames i and j are. This representation has been previously shown to effectively encode structure [40] that is useful for music retrieval applications.

Fig. Figure 3.2 shows this music matrix for the song available at https://tinyurl.com/yaurtk57. A reference segment (shown in red, spanning 0.0s to 0.8s) repeats several times later in the song (shown in green). Our music representation captures this repeating structure well.

Dance representation. Our agent is parameterized with an ordinal movement parameter k that takes one of $1, ..., K$ discrete 'states' at each step in a sequence of N 'actions'. The agent always begins in the middle $\sim \frac{K}{2}$. At each step, the agent can take one of three actions: stay at the current state k, or move to adjacent states ($k-1$ or $k+1$) without going out of bounds. We explore three ways to represent a dance.

1. State-based (**ST**). Similar to music, we define our dance matrix $\text{dance}_{\text{state}}[i, j]$ as similarity in the agent's state at time i and j: distance between the two states normalized by $(K-1)$, subtracted from 1. Similarity is 0 when the two states are the farthest possible, and 1 when they are the same.

2. Action-based (**AC**). $\text{dance}_{\text{action}}[i, j]$ is 1 when the agent takes the same action at times i and j, and 0 otherwise.

3. State + action-based (**SA**). As a combination, $\text{dance}_{\text{state+action}}$ is the average of $\text{dance}_{\text{state}}$ and $\text{dance}_{\text{action}}$.

Reasoning about tuples of states and actions (as opposed to singletons at i and j) is future work.

Objective function: aligning music and dance. We use Pearson correlation between vectorized music and dance matrices as the objective function our agent optimizes to search for 'good' dances. Pearson correlation measures the strength of linear association between the two representations, and is high when the two matrices are aligned (leading to well-synchronized dance) and low if unsynchronized.

For an $M \times M$ music matrix and $N \times N$ dance matrix (where $N =$ no. of actions), we upsample the dance matrix to $M \times M$ via nearest neighbor interpolation and then compute Pearson correlation. That is, each step in the dance corresponds to a temporal window in the input music.

In light of this objective, we can now intuitively understand our dance representations.

State-based (ST): Since this is based on distance between states, the agent is encouraged to position itself so that it revisits similar states when similar music sequences repeat. Note that this could be restrictive in the actions the agent can take or hard to optimize as it requires planning actions in advance to land near where it was when the music repeats.

Action-based (AC): Since this is based on matching actions, the agent is encouraged to take actions such that it takes the same actions when similar music sequences repeat. This has a natural analogy to human dancers who often repeat moves when the music repeats. Intuitively, this is less restrictive than ST because unlike states, actions are independent and not bound by transition constraints; recall that the agent can only move to adjacent states from a state (or stay).

Search procedure. We use Beam Search with a single beam to find the best dance sequence given the music and dance matrices, as scored by the Pearson correlation objective described earlier. We use chunks of 5 dance steps as one node in the beam. The node can take one of 3^5 values (3 action choices at each step). Specifically, we start with the first 5 steps and the corresponding music matrix (red boxes in Fig. Figure 3.3). We compute Pearson correlation with all 3^5 dance matrices, and return the best sequence for these 5 steps. Next, we set the first 5 steps to the best sequence, and search over all combinations of the next 5, *i.e.*, 3^5 sequences, each of length 10 now. See orange boxes in Fig. Figure 3.3. This continues till a sequence of length N has been found (i.e., the music ends). Our music and dance representations scale well with song length. Our search procedure scales linearly with number of steps in the dance. While its greedy nature allows the agent to dance $\sim$live with the music, it may result in worse synchronization for later parts of the song. It scales exponentially with number of actions, and we discuss approaches to overcome this in Future Work.

Baselines. We hypothesize that dances that have a balance of surprise and value will be perceived to be more creative. That is, dances where an agent moves predictably or that are

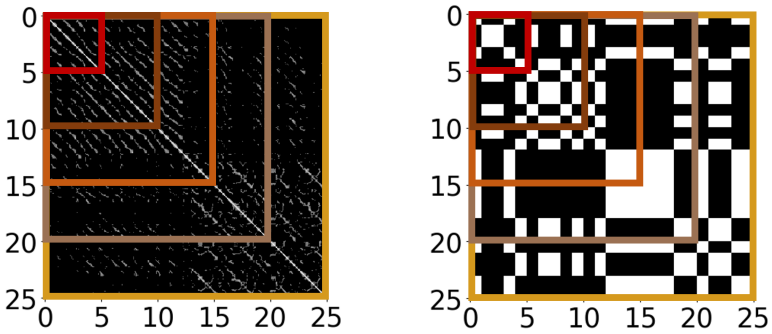


Figure 3.3: Our search procedure sequentially searches for dance sequences that result in high alignment between corresponding music (left) and dance (right) matrices. Sequential ordering shown as red to lighter shades of orange.

not synchronized with the music will be deemed less creative. This motivates our baselines:

1. Synced, sequential (SS). The agent moves sequentially from one extreme of the state space to the other till the music ends. It only moves when there is a beat in the music. The beat information is extracted using the implementation of [41] from the Madmom library. This baseline is synced, yet predictable and uninteresting.

2. Un-synced, sequential (US). The agent also moves sequentially, but ignores the beats and moves at every step. This baseline is unsynced and predictable.

3. Synced, random (SR). The agent takes a random action from its allowed actions at every beat, and stays put otherwise. This baseline is synced and quite unpredictable, so we expect this to be more interesting than SS and US.

4. Un-synced, random (UR). The agent takes a random action from its allowed actions independent of when the beat occurs. This baseline is unsynced and unpredictable.

3.5 Evaluation via human studies

We compare our approach using the 3 dance representations against the 4 baselines for 25 song snippets and values of $N \in \{25, 50, 100\}$ (no. of steps in the dance). We set the number of states an agent can be in to $K = 20$. For this experiment, we visualize the agent as a dot, with state indicating a location on a 1-D grid. We first compare approaches

with the same N, and then for the best approach, compare different Ns. We then compare our best approach to the strongest baseline using other visualizations to evaluate the role visualization plays in perception of dance creativity. For each of these settings, we show subjects on Amazon Mechanical Turk (AMT) a pair of dances and ask them: Which dance (1) goes better with the music? (2) is more surprising / unpredictable? (3) is more creative? (4) is more inspiring? Subjects can pick one of the two dances or rate them equally.

Dance representation. The 7 approaches amount to $\binom{7}{2} = 21$ pairs of dances per song per N. We showed each pair (for the same song and N) to 5 subjects on AMT. For the 25 songs and $N \in \{25, 50, 100\}$, this results in a total of 7875 comparisons. 210 unique subjects participated in this study.

See Fig. Figure 3.4. Table cells show win rate of approach in row against approach in column. Subscripts and green shades show statistical confidence levels (shown only for $> 80\%$). For example, dances from SR are found to be more creative than those from SS 61% of the times. That is, at our sample size, SR is more creative than SS with 99% confidence. Among baselines (rows 1 to 4), humans found random variants (SR, UR) to be more unpredictable (as expected), and UR to be more creative, better synchronized to music, and more inspiring than sequential variants (SS, US). UR is the best-performing baseline across metrics. We hypothesize that UR performs better than SR because the latter only moves with beats. Comments from subjects indicate that they prefer agents that also move with other features of the music. All our proposed approaches perform better than SS, US, SR baselines across metrics. AC is rated comparable to ST and SA in terms of (un)predictability. But more creative, synchronized with music, and inspiring. This may be because as discussed earlier, state-based synchronization is harder to achieve. Moreover, repetition in actions for repeating music is perhaps more common among dancers than repetition in states (location). Finally, our best approach AC is rated as more creative than the strongest baseline UR.

Number of steps. With a higher number of steps, the agent can sync to the music with

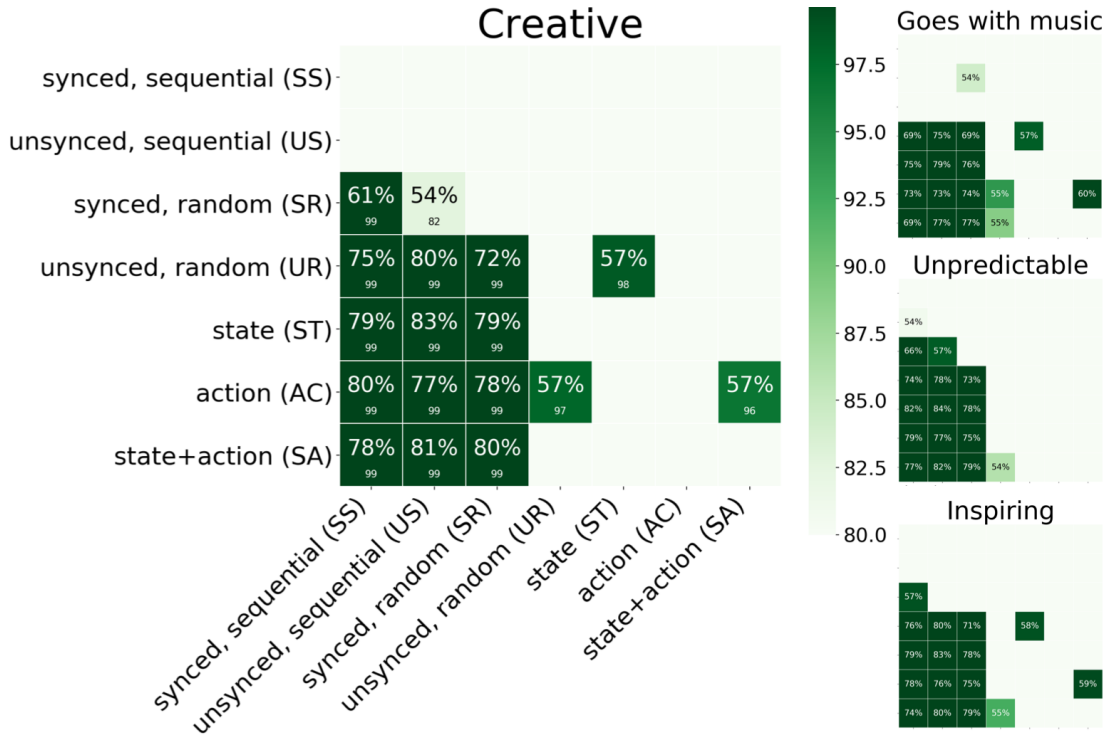


Figure 3.4: Evaluation via human studies of dances on 4 metrics – a) creativity, b) synchronization with music, c) unpredictability, and d) inspiration. Table cells show win rate of approach in row against approach in column. Shade of green and subscript shows statistical confidence (only for $> 80\%$).

higher precision. However more steps would add more "jumpiness" to the dance, which may not be desirable. We evaluate our best approach (AC) for $N \in \{25, 50, 100\}$. This gives us $\binom{3}{2} = 6$ pairs of dances per song. We showed each pair for each of the 25 songs to 5 AMT subjects; 375 pairwise comparisons from 22 unique subjects. Subjects find dances with 100 steps to be more creative than 50 and 25 steps at 99% statistical confidence, with 100 steps preferred 69% and 73% of the times respectively.

Effect of visualizations. Finally, we analyze how choice of visualization affects perception of dance creativity. We compare our best approach (AC) with the strongest baseline (UR) for 6 different visualizations including a pulsating disc, a stick figure, and collections of deforming geometric shapes. Including the dot on a 1-D grid from earlier, we have 7 pairs of dances for 25 songs and 5 evaluators; 875 comparisons from 59 unique subjects. Preference for our approach for creativity ranges from 48% to 61% across visualizations, with 2 visualizations at $<$50%. Preference on only 3 of the 7 visualizations is significant at $>$95%

confidence; all favor our approach. Interestingly, one of these visualizations corresponds to a human stick figure. Perhaps nuances in dance are more easily perceived with human-like visualizations. Example dances of our best approach (AC) for different songs, number of steps, visualizations, and song durations can be found at https://tinyurl.com/ycoz6az8.

3.6 Discussion

Our preliminary study with a simple agent gives promising indications that subjects find dances discovered using our flexible, intuitive heuristic to be creative. The next step is to train more complex agents to dance. Our search-based approach will not scale well with larger action spaces. We plan to use machine learning approaches to optimize for the music-dance alignment, so that given a new song at test time, an aligned dance sequence an be produced without an explicit search. Rather than supervised learning approaches described in Related Work which require annotated data, we will explore Reinforcement Learning (RL) using our objective function as a reward. This retains the the possibility of discovering novel dances, which is central to creativity.

CHAPTER 4

CONCLUSION

This thesis described two efforts in developing tools that can provide creative inspiration to artists. In the first work, we described an unsupervised approach to creatively stylize a word using theme-based cliparts. In the second work, we presented a general computational approach to enable a machine to generate a dance for any input music. We conducted human studies to systematically evaluate our approaches and their creative impact.

This is of course just an early step in the direction of building machines that are not only intelligent, but also understand the value of creative works. These tasks can also serve as test-beds to demonstrate intelligence of today's machines. We believe that this research does not stop here and that our approaches must be constantly evolving to account for changing times and needs of artists. Our code and infrastructure are publicly available so as to enable others to make use or build on top of our settings.

The field of creative technologies is fast evolving and very promising. We believe this to be a step towards bridging the gap between science and art so that researchers and artists may work together towards building a smarter and better world. While there is still a lot more to be done in this space, we optimistically look forward to an exciting new era of machines helping humans think better, collaborate better, approach problems from new perspectives, express (and even understand) each other better and in general enhance human creativity.

REFERENCES

[1] A. Frutiger, "Signs and symbols," *Their design and meaning*, 1989.

[2] D. Schmandt-Besserat, "The evolution of writing," *International Encyclopedia of the Social and Behavioral Sciences: Second Edition*, 2015.

[3] M. Shiojiri and Y. Nakatani, "Visual language communication system with multiple pictograms converted from weblog texts," in *IASDR*, 2013.

[4] T. H. Clawson, J. Leafman, G. M. Nehrenz Sr, and S. Kimmer, "Using pictograms for communication," *Military medicine*, 2012.

[5] T. Takasaki and Y. Mori, "Design and development of a pictogram communication system for children around the world," in *Intercultural collaboration*, 2007.

[6] M. Cunha, P. Martins, João, and P. Machado, "How shell and horn make a unicorn: Experimenting with visual blending in emoji," in *ICCC*, 2018.

[7] P. Martins, T. Urbancic, S. Pollak, N. Lavrac, and A. Cardoso, "The good, the bad, and the aha! blends," in *ICCC*, 2015.

[8] P. Xiao and S. Linkola, "Vismantic: Meaning-making with images," in *ICCC*, 2015.

[9] G. Atarsaikhan, B. K. Iwana, and S. Uchida, "Contained neural style transfer for decorated logo generation," in *IAPR DAS*, 2018.

[10] A. Sage, E. Agustsson, R. Timofte, and L. Van Gool, "Logo synthesis and manipulation with clustered generative adversarial networks," in *CVPR*, 2018.

[11] D. Ha and D. Eck, "A neural representation of sketch drawings," in *ICLR*, 2018.

[12] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *CVPR*, 2017.

[13] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *ICCV*, 2017.

[14] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[15] S. Azadi, M. Fisher, V. G. Kim, Z. Wang, E. Shechtman, and T. Darrell, "Multi-content gan for few-shot font style transfer," in *CVPR*, 2018.

[16] N. D. Campbell and J. Kautz, "Learning a manifold of fonts," *Trans. on Graphics*, 2014.

[17] T. Martins, J. Correia, E. Costa, and P. Machado, "Evotype: Towards the evolution of type stencils," in *International Conference on Computational Intelligence in Music, Sound, Art and Design*, 2018.

[18] G. Atarsaikhan, B. K. Iwana, A. Narusawa, K. Yanai, and S. Uchida, "Neural font style transfer," in *ICDAR*, 2017.

[19] D. H. Ballard, "Modular learning in neural networks.," in *AAAI*, 1987.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[21] G. Rawlinson, "The significance of letter position in word recognition," *IEEE Aerospace and Electronic Systems Magazine*, 2007.

[22] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS*, 2013.

[23] J. Wieting, M. Bansal, K. Gimpel, and K. Livescu, "Towards universal paraphrastic sentence embeddings," in *ICLR*, 2016.

[24] Y. Adi, E. Kermany, Y. Belinkov, O. Lavi, and Y. Goldberg, "Fine-grained analysis of sentence embeddings using auxiliary prediction tasks," in *ICLR*, 2017.

[25] J. Schmidhuber, "Formal theory of creativity, fun, and intrinsic motivation (1990–2010)," *IEEE Transactions on Autonomous Mental Development*, 2010.

[26] T. Appenzeller, *Evolution or revolution?* 1998.

[27] S. Brown, M. J. Martinez, and L. M. Parsons, "The neural basis of human dance," *Cerebral cortex*, 2006.

[28] I. Infantino, A. Augello, A. Manfré, G. Pilato, and F. Vella, "Robodanza: Live performances of a creative dancing humanoid," in *ICCC*, 2016.

[29] A. Augello, E. Cipolla, I. Infantino, A. Manfre, G. Pilato, and F. Vella, "Creative robot dance with variational encoder," *arXiv:1707.01489*, 2017.

[30] N. Yalta, S. Watanabe, K. Nakadai, and T. Ogata, "Weakly-supervised deep recurrent neural networks for basic dance step generation," in *IJCNN*, 2019.

[31] A. Manfré, I. Infantino, A. Augello, G. Pilato, and F. Vella, "Learning by demonstration for a dancing robot within a computational creativity framework," in *IRC*, 2017.

[32] H.-Y. Lee, X. Yang, M.-Y. Liu, T.-C. Wang, Y.-D. Lu, M.-H. Yang, and J. Kautz, "Dancing to music," in *NeurIPS*, 2019.

[33] J. Lee, S. Kim, and K. Lee, "Listen to dance: Music-driven choreography generation using autoregressive encoder-decoder network," *arXiv:1811.00818*, 2018.

[34] W. Zhuang, C. Wang, S. Xia, J. Chai, and Y. Wang, "Music2dance: Music-driven dance generation using wavenet," *arXiv:2002.03761*, 2020.

[35] M. Pettee, C. Shimmin, D. Duhaime, and I. Vidrin, "Beyond imitation: Generative and variational choreography via machine learning," *arXiv:1907.05297*, 2019.

[36] K. Carlson, T. Schiphorst, and P. Pasquier, "Scuddle: Generating movement catalysts for computer-aided choreography," in *ICCC*, 2011.

[37] M. Jacob and B. Magerko, "Interaction-based authoring for scalable co-creative agents," in *ICCC*, 2015.

[38] T. Tang, J. Jia, and H. Mao, "Dance with melody: An lstm-autoencoder approach to music-oriented dance synthesis," in *ACM MM*, 2018.

[39] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "Librosa: Audio and music signal analysis in python," in *SciPy*, 2015.

[40] J. T. Foote and M. L. Cooper, "Media segmentation using self-similarity decomposition," in *Storage and Retrieval for Media Databases*, 2003.

[41] F. Krebs, S. Böck, and G. Widmer, "An efficient state-space model for joint tempo and meter tracking," in *ISMIR*, 2015.